

Rerouting flows when links fail*

Jannik Matuschke[†] S. Thomas McCormick[‡]
 Gianpaolo Oriolo[§]

Abstract

We introduce and investigate *reroutable flows*, a robust version of network flows in which link failures can be mitigated by rerouting the affected flow. Given a capacitated network, a path flow is reroutable if after failure of an arbitrary arc, we can reroute the interrupted flow from the tail of that arc to the sink, without modifying the flow that is not affected by the failure. Similar types of restoration, which are often termed “local”, were previously investigated in the context of network design, such as min-cost capacity planning. In this paper, our interest is in computing maximum flows under this robustness assumption. An important new feature of our model, distinguishing it from existing max robust flow models, is that no flow can get lost in the network.

We also study a tightening of reroutable flows, called *strictly reroutable flows*, making more restrictive assumptions on the capacities available for rerouting. For both variants, we devise a reroutable-flow equivalent of an s - t -cut and show that the corresponding max flow/min cut gap is bounded by 2. It turns out that a strictly reroutable flow of maximum value can be found using a compact LP formulation, whereas the problem of finding a maximum reroutable flow is NP -hard, even when all capacities are in $\{1, 2\}$. However, the tightening can be used to get a 2-approximation for reroutable flows. This ratio is tight in general networks, but we show that in the case of unit capacities, every reroutable flow can be transformed into a strictly reroutable flow of same value. While it is NP -hard to compute a maximal integral flow even for unit capacities, we devise a surprisingly simple combinatorial algorithm that finds a half-integral strictly reroutable flow of value 1, or certifies that no such solutions exists. Finally, we also give a hardness result for the case of multiple arc failures.

1 Introduction

Network infrastructures for transportation, communication, or energy transmission are an important backbone of our society. However, they are also prone to failure or intentional sabotage, and in such cases it is desirable to quickly

*An extended abstract of this work appears in the proceedings of ICALP 2017.

[†]TUM School of Management and Department of Mathematics, Technische Universität München

[‡]Sauder School of Business, University of British Columbia

[§]Dipartimento di Ingegneria Civile e Ingegneria Informatica, Università di Roma “Tor Vergata”

recover the service provided through the network. A crucial frequent requirement of actual network restoration techniques is that restoration is handled locally [16]. As a motivating example, consider a communication network in which data packets are routed along paths. When a link in the network fails, it is desirable to only reroute the traffic that is actually affected by the failure, i.e., those paths that traverse the failing link, without changing or rerouting any part of the flow that is not affected by the failure. Note that arbitrary rearrangement of the flow after a failure is in general more powerful, but it is both undesirable to interrupt customer service and hard to do so reliably and safely [11, 20].

To cope with such a situation, we introduce the concept of reroutable network flows: A flow on s - t -paths is *reroutable* if after failure of any arc $\bar{a} = (\bar{v}, \bar{w})$ in the network, we can reroute all flow that was traversing \bar{a} from \bar{v} to the sink t , while not changing any flow that was not affected by the interruption. Similar concepts were previously discussed in a few other papers [8, 9, 22, 23], but with an emphasis on network design issues, e.g., minimizing the cost of the installed capacity. In contrast, our interest is in computing maximum flows (but we point out that a potential application are feasibility/separation subroutines for capacity reservation). Note that in this setting, we cannot simply send a standard maximum flow, as we need to leave space for rerouting. Before we discuss our findings and better relate them to existing literature, let us formalize the definition of our model.

Network flows Let $D = (V, A)$ be a digraph with source $s \in V$, a sink $t \in V$ and arc capacities $u \in \mathbb{R}_+^A$. Let $\mathcal{P} \subseteq 2^A$ be the set of simple¹ s - t -paths in D . For arcs $a, \bar{a} \in A$, define

$$\mathcal{P}_a := \{P \in \mathcal{P} : a \in P\} \quad \text{and} \quad \mathcal{P}_{\bar{a} \rightarrow a} := \{P \in \mathcal{P} : a, \bar{a} \in P, \bar{a} \prec_P a\},$$

where $\bar{a} \prec_P a$ means that P traverses \bar{a} before a . An s - t -flow is a vector $x \in \mathbb{R}_+^{\mathcal{P}}$ that assigns a flow value $x(P) \geq 0$ to each $P \in \mathcal{P}$ such that the arc flow values $x(a) := \sum_{P \in \mathcal{P}_a} x(P)$ fulfill the capacity constraint $x(a) \leq u(a)$ for all $a \in A$. The value of a flow x is $\text{val}(x) := \sum_{P \in \mathcal{P}} x(P)$.

Reroutable flows Let x be an s - t -flow. If an arc $\bar{a} = (\bar{v}, \bar{w}) \in A$ fails, all flow on paths containing the failing arc gets interrupted when it reaches \bar{v} . For any $a \in A \setminus \{\bar{a}\}$, we define the *available capacity* of a after failure of \bar{a} by

$$\bar{u}_{x, \bar{a}}(a) := u(a) - \sum_{P \in \mathcal{P}_a \setminus \mathcal{P}_{\bar{a} \rightarrow a}} x(P).$$

A *rerouting* of x for the failing arc \bar{a} is a \bar{v} - t -flow $x_{\bar{a}}$ of value $x(\bar{a})$ in $(V, A \setminus \{\bar{a}\})$ with capacities $\bar{u}_{x, \bar{a}}$. The flow x is *reroutable* if for every failing arc $\bar{a} \in A$ there is a rerouting $x_{\bar{a}}$ of x .

Strictly reroutable flows A rerouting $x_{\bar{a}}$ of a flow x for a failing arc \bar{a} is *strict* if $x_{\bar{a}}(a) \leq \bar{u}_{x, \bar{a}}(a) := u(a) - x(a)$ for every $a \in A \setminus \{\bar{a}\}$. We say that x is *strictly reroutable* if for every failing arc $\bar{a} \in A$ there is a strict rerouting of x .

¹All our results also work for the case that \mathcal{P} contains non-simple paths, but we restrict to simple paths for ease of notation.

Strictly reroutable flows are both a helpful tool for computing reroutable flows and interesting in their own right, in situations where more conservative assumptions have to be made on the capacities available for rerouting. A natural question is what is the maximum flow value that can be sent by a (strictly) reroutable flow in a given network. We denote the corresponding optimization problem as MAX RF and MAX SRF, respectively.

Path flows vs. arc flows Our reroutable flow model is defined for path flows, which is a common assumption in many robust flow models [2, 4, 15, 19]. It is well-known that network flows allow for two different representations: By specifying a value for each path, as is done in this paper, or by specifying the flow on each arc and requiring flow conservation at each node of the network. Arc flow values can easily be obtained from a given path flow. Conversely, every arc flow can be decomposed into flow on paths, but in general this decomposition is not unique; see, e.g., [3]. There are flow problems where the path decomposition does not play a role, e.g., the maximum flow problem or the minimum cost flow problem, and others where it does, e.g., robust flows [4], length-bounded flows [5], or the computation of earliest arrival flows [21]. For our model, it turns out that MAX RF falls into the first category, whereas MAX SRF falls into the second.

1.1 Our results

Complexity of the problems (Sections 2.1 and 2.2) We observe that MAX SRF can be solved in polynomial time by formulating it as a linear program. In contrast, MAX RF is *NP*-hard, even when $u(a) \in \{1, 2\}$ for all $a \in A$. On the positive side, by showing that the maximum value of a reroutable flow is at most twice as large as the maximum value of a strictly reroutable flow, we obtain a 2-approximation for MAX RF for arbitrary capacities. The problem can further be solved exactly in unit capacity networks (see below).

Max flow/min cut gap (Section 2.3) Max flow/min cut results play a central role in network flow theory. We devise a combinatorial upper bound for the maximum reroutable flow value, called *R-cut*, and prove that the corresponding flow/cut gap for both reroutable and strictly reroutable flows is bounded by 2. In fact, our proof is constructive and provides a combinatorial 2-approximation algorithm for the minimum capacity R-cut problem.

Unit capacity networks (Section 3) We consider the case of unit capacities. It turns out that in this case, MAX RF and MAX SRF are equivalent. Our proof is based on a careful uncrossing argument that allows to transform any reroutable flow into a strictly reroutable flow.

Computing (half-)integral solutions (Section 4) A common property of many flow problems is the existence of an integral optimal solution when capacities are integral. In the case of reroutable flows, this property does not hold. In fact, if we require flow to be integral, the problem becomes *NP*-hard, even for sending a single unit of flow in a unit capacity network. However, for this special case, we devise a simple combinatorial algorithm that computes a

half-integral solution or certifies that no flow of value 1 exists. Via our max flow/min cut analysis we also show how to compute 2-approximate half-integral solutions.

Multiple arc failures (Section 5.3) We consider the natural generalization of our problems to multiple simultaneous arc-failures. We show that in this case both variants of the problem are *NP*-hard, even when only two arcs can fail and all arcs have unit capacity. All hardness results in this paper are based on reduction from an intermediary problem, called FORBIDDEN PAIRS *s-t*-PATH. They are therefore grouped together in Section 5.

1.2 Related work

As we already pointed out above, “local” rerouting schemes, i.e., schemes that only change flow affected by the failure, have been investigated in network design. A routing scheme in which flow has to be sent along arc-disjoint paths was investigated in [8], see also [23]. The problem of finding a local rerouting from the tail to the head of a failed arc was investigated in [9] and [22]. However, in all these papers the focus was on min-cost capacity planning.

Concepts that deal with the maximization of flow subject to robustness constraints commonly fall under the moniker of *robust flows*. Aggarwal and Orlin [2] studied *k-route flows*. Such a flow is a conic combination of elementary flows, each of which consists of a uniform flow along k disjoint paths. Because of this structure, the failure of any arc can only destroy a $1/k$ fraction of the flow. A maximum k -route flow can be computed in polynomial time by means of a parametric max flow problem. Another classic model is the *maximum robust flow problem*: Here, the goal is to find a path flow that maximizes the surviving flow after a worst-case failure of k arcs. Aneja et al. [4] showed that for $k = 1$ both an optimal fractional and an optimal integral solution can be found in polynomial time. If k is not bounded by a constant the problem is *NP*-hard [12], but the complexity for any constant value $k \geq 2$ is open. Bertsimas et al. [6] provide an $\Omega(1/k)$ -approximation algorithm for the maximum robust flow. Robust flows are closely related to *network flow interdiction*, which takes a dual perspective: The goal is to find a subset of arcs whose removal minimizes the maximum flow value in the remaining network; see the recent article by Chestnut and Zenklusen [10] for an up-to-date overview of this topic.

To the best of our knowledge, the only other flow maximization model that allows for adjustment after the failure are *adaptive flows*, first introduced by Bertsimas et al. [7]: In the first step, an arc flow is specified. After failure of k arcs, a new flow is sent, with the flow value on every arc being bounded by the original flow value. Note that adaptive flows differ from reroutable flows in two important aspects: Adaptive flows allow flow to be ‘lost’ (the flow value after the failure is lower than the original flow value), whereas in reroutable flows all flow has to reach the sink. Furthermore, adaptive flows can reconfigure the flow in the entire network, whereas in reroutable flows, only the flow affected by the failure can be rerouted.

Another model closely related to reroutable flows is the *online replacement path* problem (ORP) introduced by Adjiashvili et al. [1]. The ORP is a generalization of the shortest path problem: Given a digraph with costs on the arcs, we have to specify an *s-t*-path. Along the path, we may encounter a failing arc

$\bar{a} = \{\bar{v}, \bar{w}\}$, and we have to find a replacement path from \bar{v} to t avoiding \bar{a} . The goal is to minimize the total traveled distance, assuming \bar{a} is chosen by an adversary. Adjashvili et al. [1] show that the ORP can be solved in polynomial time, even when a constant number of arcs fail.

2 LP formulation, approximation, and max flow/min cut

In this section, we discuss the complexity of the two problems and provide bounds on the gap between MAX RF and MAX SRF. We also introduce an analogue to minimum cuts for reroutable flows and bound the corresponding duality gap. At the end of the section, we show that all our bounds are tight.

2.1 Complexity of Max RF and Max SRF

We now consider an LP formulation for MAX SRF. For $\bar{a} \in A$, let $\mathcal{R}(\bar{a})$ be the set of all $\text{tail}(\bar{a})$ - t -paths in $(V, A \setminus \{\bar{a}\})$, which are exactly the paths that a rerouting for failing arc \bar{a} can use.

$$\begin{aligned}
[\text{LP}_{\text{strict}}] \quad & \max \quad \sum_{P \in \mathcal{P}} x(P) \\
\text{s.t.} \quad & \sum_{P \in \mathcal{P}_a} x(P) + \sum_{R \in \mathcal{R}(\bar{a}) : a \in R} x_{\bar{a}}(R) \leq u(a) \quad \forall a, \bar{a} \in A \\
& \sum_{P \in \mathcal{P}_{\bar{a}}} x(P) - \sum_{R \in \mathcal{R}(\bar{a})} x_{\bar{a}}(R) = 0 \quad \forall \bar{a} \in A \\
& x, x_{\bar{a}} \geq 0 \quad \forall \bar{a} \in A
\end{aligned}$$

The first set of constraints bound the capacities for each rerouting; note in particular that for $\bar{a} = a$, the second term becomes 0, ensuring $x(a) \leq u(a)$ for all $a \in A$. The second set of constraints ensures that the rerouting flow $x_{\bar{a}}$ has value $x(\bar{a})$.

For our discussion, it will also be useful to consider the dual of $[\text{LP}_{\text{strict}}]$. We introduce dual variables $y_{\bar{a}}(a)$ for every $a, \bar{a} \in A$ and $z(\bar{a})$ for every $\bar{a} \in A$.

$$\begin{aligned}
[\text{D-LP}_{\text{strict}}] \quad & \min \quad \sum_{a \in A} \sum_{\bar{a} \in A} u(a) y_{\bar{a}}(a) \\
\text{s.t.} \quad & \sum_{a \in P} \left(z(a) + \sum_{\bar{a} \in A} y_{\bar{a}}(a) \right) \geq 1 \quad \forall P \in \mathcal{P} \\
& \sum_{a \in P} y_{\bar{a}}(a) - z(\bar{a}) \geq 0 \quad \forall \bar{a} \in A, P \in \mathcal{R}(\bar{a}) \\
& y_{\bar{a}}(a) \geq 0 \quad \forall a, \bar{a} \in A
\end{aligned}$$

Although $[\text{LP}_{\text{strict}}]$ has an exponential number of variables, it can be solved in polynomial time via dual separation.

Theorem 1. MAX SRF can be solved in polynomial time.

Proof. We observe that the separation problem to $[\text{D-LP}_{\text{strict}}]$ can be solved by a sequence of shortest path computations: Consider a point y, z . A violation of the first set of constraints can be detected by computing the shortest s - t -paths with respect to arc costs $c_{y,z}(a) := z(a) + \sum_{\bar{a} \in A} y_{\bar{a}}(a)$. If the shortest path has cost smaller than 1, it corresponds to a violated inequality, otherwise y, z fulfills the inequalities. A violation of the second set of constraints can be detected by computing a shortest $\text{tail}(\bar{a})$ - t -path for every $\bar{a} \in A$ with respect to arc costs $y_{\bar{a}}(a)$. If the shortest path for some \bar{a} has cost smaller than $z(\bar{a})$ the constraint is violated for that path. Otherwise y, z fulfills the inequalities.

Hence we can obtain an optimal solution for $[\text{D-LP}_{\text{strict}}]$ using the equivalence of optimization and separation. By restricting $[\text{LP}_{\text{strict}}]$ to the paths corresponding to inequalities generated during the solution of the dual, we obtain a primal solution of equal value. \square

Remark 2. *An alternative way to obtain a polynomial algorithm for MAX SRF is to observe that the capacities $\bar{u}_x(a) = u(a) - x(a)$ available for strict rerouting only depend on the arc flow values. Hence, we can formulate MAX SRF as a compact LP with arc flow variables. Then any path decomposition of the resulting arc flow is a maximum strictly reroutable flow.*

For the special case of unit capacity networks, we show in Section 3 that an optimal solution to $[\text{LP}_{\text{strict}}]$ is also optimal for MAX RF.

Theorem 3. *For $u \equiv 1$, MAX RF can be solved in polynomial time.*

An LP formulation for MAX RF can be obtained by replacing the term $\sum_{P \in \mathcal{P}_a} x(P)$ by $\sum_{P \in \mathcal{P}_{\bar{a} \rightarrow a}} x(P)$ in the capacity constraints of $[\text{LP}_{\text{strict}}]$. Unfortunately, this modification prevents the dual separation approach from working. In fact, it turns out that MAX RF is hard as soon as two different capacities occur. The proof of this result is discussed in Section 5.1.

Theorem 4. *MAX RF is NP-hard, even when $u(a) \in \{1, 2\}$ for all $a \in A$.*

2.2 Reroutable flows vs. strictly reroutable flows

As MAX SRF is a tightening of MAX RF, the optimal value of the former is at most that of the latter. We show that the gap between the two values cannot be larger than 2. As we can compute maximum strictly reroutable flows, we obtain a 2-approximation for MAX RF.

Lemma 5. *Let x be an s - t -flow. If x is strictly reroutable, then x is reroutable. If x is reroutable, then $\frac{1}{2}x$ is strictly reroutable.*

Proof. The first statement follows from the fact that $\bar{u}_{x,\bar{a}}(a) \geq \bar{u}_x(a)$ for all $\bar{a}, a \in A \setminus \{\bar{a}\}$. For the proof of the second statement, assume x is reroutable and consider any $\bar{a} = (\bar{v}, \bar{w}) \in A$. Let $\bar{x}_{\bar{a}}$ be the rerouting of x in case of failure of \bar{a} . Now observe that $\frac{1}{2}\bar{x}_{\bar{a}}$ is a strict rerouting of $\frac{1}{2}x$ in case of failure of \bar{a} , because it is a \bar{v} - t -flow of value $\frac{1}{2}x(\bar{a})$ and

$$\frac{1}{2}\bar{x}_{\bar{a}}(a) \leq \frac{1}{2} \left(u(a) - \sum_{P \in \mathcal{P}_a \setminus \mathcal{P}_{\bar{a} \rightarrow a}} x(P) \right) \leq u(a) - \frac{1}{2} \sum_{P \in \mathcal{P}_a} x(P) = \bar{u}_{x/2}(a),$$

where the second inequality follows from $\sum_{P \in \mathcal{P}_{\bar{a} \rightarrow a}} x(P) \leq u(a)$. \square

Corollary 6. *There is a 2-approximation algorithm for MAX RF.*

2.3 Max flow/min cut gap for reroutable flows

An s - t -cut is a set of arcs that intersects every s - t -path. Its *capacity* is the sum of capacities of its arcs. A fundamental result in network flow theory is that the value of a maximum s - t -flow is equal to the capacity of a minimum s - t -cut. This result has been successfully generalized to many variants of network flows, such as abstract flows [17] or flows over time [13]. However, in other cases, such as multicommodity flows, the equality does not hold and instead, researchers investigate the worst case ratio between maximum flow and minimum cut; see, e.g., [18].

We present a counterpart to an s - t -cut for reroutable flows. It turns out that max flow and min cut are not necessarily equal and we give a tight bound on the corresponding max flow/min cut gap. An R -cut is a set of arcs $R \subseteq A$ together with a collection of cuts $(C_a)_{a \in R}$, where each C_a is a tail(a)- t -cut containing a . We denote $(R, (C_a)_{a \in R})$ by (R, C) for short. The capacity of the R -cut (R, C) is

$$\text{cap}(R, C) := \phi(R, C) + \sum_{a \in R} u(C_a \setminus \{a\}),$$

where $\phi(R, C)$ is the capacity of a minimum s - t -cut in $(V, A \setminus \cup_{a \in R} C_a)$.

The intuition behind this definition is the following: For every $a \in R$, all flow that crossed the cut C_a must cross the $C_a \setminus \{a\}$ if a fails. If a flow path does not cross any cut in C_a , then it crosses the minimum s - t -cut in $(V, A \setminus \cup_{a \in R} C_a)$. Therefore the capacity of an R -cut is an upper bound on the value of any reroutable flow.

Lemma 7. $\text{val}(x) \leq \text{cap}(R, C)$ for any reroutable flow x and any R -cut (R, C) .

Proof. Let $a \in R$. As C_a is a tail(a)- t -cut, $x(a)$ units of flow have to be rerouted across the arcs in $C_a \setminus \{a\}$ when a fails. Therefore

$$x(a) \leq \sum_{a' \in C_a \setminus \{a\}} \bar{u}_{x,a}(a') \leq u(C_a \setminus \{a\}) - \sum_{P \in \mathcal{P}: P \cap C_a \neq \emptyset, a \notin P} x(P).$$

This implies $\sum_{P \in \mathcal{P}: P \cap C_a \neq \emptyset} x(P) \leq u(C_a \setminus \{a\})$ for all $a \in R$. Now let S be a minimum s - t -cut in $(V, A \setminus \cup_{a \in R} C_a)$. Then

$$\begin{aligned} \text{val}(x) &= \sum_{P \in \mathcal{P}} x(P) \leq \sum_{P \in \mathcal{P}: P \cap S \neq \emptyset} x(P) + \sum_{a \in R} \sum_{P \in \mathcal{P}: P \cap C_a \neq \emptyset} x(P) \\ &\leq \sum_{a \in S} u(a) + \sum_{a \in R} u(C_a \setminus \{a\}) = \text{cap}(R, C). \end{aligned} \quad \square$$

At the end of this section, we further show that R -cuts correspond to integral solutions to $[D\text{-LP}_{\text{strict}}]$. We now give a constructive proof bounding the duality gap between maximum strictly reroutable flow and minimum R -cut (or, equivalently, the integrality gap of the dual LP). In Section 2.4 we give an example showing that the bound is tight.

Theorem 8. Let x be a strictly reroutable flow of maximum value and let (R, C) be an R -cut of minimum capacity. Then $\text{val}(x) \geq \frac{1}{2} \text{cap}(R, C)$.

Proof. For $a \in A$, let C_a be minimum tail(a)- t -cut in D containing a and define $u'(a) := \min\{u(a), u(C_a \setminus \{a\})\}$. Let C' be a minimum s - t -cut in D with respect

to the capacities u' and let x' be a corresponding maximum flow. Now define $R := \{a \in C' : u'(a) < u(a)\}$. Observe that R and $(C_a)_{a \in R}$ define an R-cut and that $\phi(R, C) \leq u(C' \setminus R)$. We obtain

$$\text{cap}(R, C) \leq \sum_{a \in C' \setminus R} u(a) + \sum_{a \in R} u(C_a \setminus \{a\}) = \sum_{a \in C'} u'(a) = \text{val}(x').$$

Now let $x := x'/2$. It is sufficient to show that x is a strictly reroutable flow. By contradiction assume that there is $\bar{a} \in A$ for which there is no strict rerouting of x . By the max flow/min cut theorem, there must be a $\text{tail}(\bar{a})$ - t -cut \bar{C} in $(V, A \setminus \{\bar{a}\})$ with $\sum_{a \in \bar{C}} \bar{u}_x(a) < x(\bar{a})$. Note that $x(a) \leq u'(a)/2 \leq u(a)/2$ for every $a \in A$ by construction of x . Thus

$$\frac{1}{2} \sum_{a \in \bar{C}} u(a) \leq \sum_{a \in \bar{C}} (u(a) - x(a)) < x(\bar{a}) \leq \frac{1}{2} u'(\bar{a}) \leq \frac{1}{2} u(C_{\bar{a}} \setminus \{\bar{a}\}).$$

However, this implies that $\bar{C} \cup \{\bar{a}\}$ is a smaller $\text{tail}(\bar{a})$ - t -cut than $C_{\bar{a}}$, a contradiction. \square

Computing a minimum capacity R-cut Let us denote the problem of finding an R-cut of minimum capacity by MIN R-CUT. The proof of Theorem 8 describes how to compute a 2-approximate solution to this problem.

Corollary 9. *There is a 2-approximation algorithm for MIN R-CUT.*

Proof. For every $a \in A$ the capacity $u'(a)$ and the corresponding cut C_a can be computed by a standard minimum s - t -cut computation. Given the values of u' , also C' can be computed by a min cut computation. \square

2.3.1 R-cuts and integral dual solutions

As mentioned above, R-cuts correspond to integral solutions of $[\text{D-LP}_{\text{strict}}]$. We give a formal argument of this correspondence.

Given an R-cut (R, C) , let C^* be a minimum s - t -cut in $(V, A \setminus \bigcup_{a \in R} C_a)$. We set $z(\bar{a}) := 1$ for every $\bar{a} \in R$, $y_{\bar{a}}(a) := 1$ for every $a \in C_{\bar{a}}$, and $y_a(a) := 1$ for every $a \in C^*$. All other variables are set to 0. It is easy to check that y, z corresponds to a feasible solution to $[\text{D-LP}_{\text{strict}}]$ with objective value $\text{cap}(R, C)$.

Conversely, consider an integral dual solution y, z . Note that we can assume that all variables take values in $\{0, 1\}$: If $z(a) < 0$ we can set it to 0 without losing feasibility. If any variable takes a value larger than 1, we can reduce it to 1 without losing feasibility. Let $R := \{\bar{a} \in A : z(\bar{a}) = 1\}$. By the second set of constraints, for every $\bar{a} \in R$, every $\text{tail}(\bar{a})$ - t path must be covered, i.e., there must be a $\text{tail}(\bar{a})$ - t -cut $C_{\bar{a}}$ with $y_{\bar{a}}(a) = 1$ for all $a \in C_{\bar{a}} \setminus \{\bar{a}\}$. By the first set of constraints, for every $P \in \mathcal{P}$ there is an $a \in P$ such that either $z(a) = 1$ or $y_{\bar{a}}(a) = 1$ for some $\bar{a} \in A$. Thus the support of y contains an s - t -cut in $(V, A \setminus \bigcup_{a \in R} C_a)$. We conclude that $\text{cap}(R, C)$ is at most the objective value of the solution y, z .

2.4 Summary of the bounds and tightness

Putting the bounds from Lemma 5 and Theorem 8 together, we obtain the following corollary.

Corollary 10. *Let (R, C) be a minimum capacity R -cut and let x_{RF} and x_{SRF} be maximal reroutable and strictly reroutable flows, respectively. Then*

$$\text{val}(x_{\text{RF}}) \leq \text{cap}(R, C) \leq 2 \text{val}(x_{\text{SRF}}) \leq 2 \text{val}(x_{\text{RF}}).$$

The following gadget will be useful throughout the paper in order to construct examples and reductions.

Backup links A *backup link* from v to w is a v - w -path (a', a'') of length 2 in which the intermediate node is incident only to the two arcs of the path and $u(a') := u(a'') := \max_{a \in A} u(a)$. Note that $x(a') = x(a'') = 0$ for any reroutable flow, because when a'' fails, there is no $\text{tail}(a'')$ - t -path for rerouting the flow on that arc. A *bidirected* backup link between v and w consists of two distinct backup links, one from v to w and one from w to v .

Tightness of bounds The network depicted in Fig. 1 shows that the bounds given in Lemma 5 and Theorem 8 are tight. Note that any reroutable flow is completely determined by the value $x(P)$ it sends along the path $P := \{a_1, a_2, a_3\}$, as backup links can only be used for rerouting.

1. For the bound on gap between max reroutable flow and max strictly reroutable flow, we set the capacities $u(a_1) = 2$ and $u(a) = 1$ for all $a \in A \setminus \{a_1\}$. We show that $x(P) = 1$ defines a reroutable flow. Failure of a_2 is not a problem, as $\text{tail}(a_2)$ has a backup link to t . If a_3 fails, flow can use the backup link to s and traverse a_1 , as $\bar{u}_{x, a_3}(a_1) = 2 - x(a_1) = 1$, to reach the backup link from $\text{tail}(a_2)$ to t . If a_1 fails, flow can be rerouted from s using the backup link and the arc a_3 , as $a_1 \prec_P a_3$ and thus $\bar{u}_{x, a_1}(a_3) = 1$. However, in a strictly reroutable flow, this last rerouting is no longer possible, as $\bar{u}_x(a_3) = 1 - x(a_3)$. Hence, $1 \geq x(a_1) + x(a_3) = 2x(P)$ for any strictly reroutable flow x . The maximum strictly reroutable flow value therefore is $1/2$.
2. For the bound on the flow/cut gap, set all capacities to 1. Then both the maximum reroutable flow value and the maximum strictly reroutable flow value are $1/2$. Consider any R -cut (R, C) . As capacities are integral, $\text{cap}(R, C) \geq 1$. An R -cut with $\text{cap}(R, C) = 1$ is, e.g., $R := \{a_3\}$ and $C_{a_3} := \{a_1, a_2\}$.

Note that Fig. 1 also shows that optimal solutions to both MAX RF and MAX SRF can be fractional, even when capacities are integral.

Remark 11. *Note that the worst-case for the bounds in Corollary 10 cannot be attained simultaneously, i.e., in any given instance either the max flow/min cut gap or the gap between reroutable and strictly reroutable flow has to be significantly smaller than 2—in fact, at least one of them has to be within $\sqrt{2}$.*

3 Unit capacity networks

Throughout this section, we assume $u \equiv 1$. We will show that in this case, any reroutable flow can be transformed into a strictly reroutable flow of the same value. While this closes the gap between the two reroutable flow variants, note

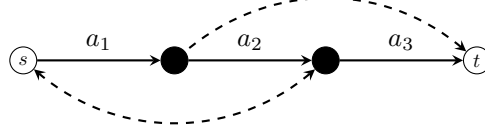


Figure 1: Example showing that the bounds given in Lemma 5 and Theorem 8 are tight. Dashed arcs correspond to (bidirected) backup links, which can only be used for rerouting. When all arcs have unit capacities, the maximum (strictly) reroutable flow has a value of $1/2$. When changing the capacity of a_1 to 2, the maximum reroutable flow value increases to 1, whereas the maximum strictly reroutable flow value remains $1/2$. The minimum R-cut capacity is 1 in both cases.

that the flow/cut gap can still be 2 in unit capacity networks, as can be seen in Fig. 1. We start by giving an alternative characterization for strictly reroutable flows in unit capacity networks.

Cuts separating t For $S \subseteq V$, let $\delta^+(S) := \{a \in A : \text{tail}(a) \in S, \text{head}(a) \in V \setminus S\}$ denote the cut induced by S . We define $\mathcal{S} := \{S \subset V \setminus \{t\} : S \neq \emptyset\}$ and let $\mathcal{C} := \{\delta^+(S) : S \in \mathcal{S}\}$ be the set of t -separating cuts. W.l.o.g. we assume $\delta^+(S) \neq \emptyset$ for all $S \in \mathcal{S}$, as no vertex in a set S with $\delta^+(S) = \emptyset$ can be on an s - t -path.

Lemma 12. *Let x be an s - t -flow for capacities $u \equiv 1$. Then x is strictly reroutable if and only if $\sum_{a \in C} (1 - x(a)) \geq 1$ for all $C \in \mathcal{C}$.*

Proof. We first show sufficiency of the condition. By contradiction assume that $\sum_{a \in C} (1 - x(a)) \geq 1$ for all $C \in \mathcal{C}$ but x is not strictly reroutable. Because x is not strictly reroutable, there must be an arc $\bar{a} \in A$ such that there is no rerouting of x for \bar{a} . This means that the maximum flow value that can be sent in $(V, A \setminus \{\bar{a}\})$ with capacities \bar{u}_x from $\bar{v} := \text{tail}(\bar{a})$ to t is strictly smaller than $x(\bar{a})$. By max flow/min cut, this implies there is a \bar{v} - t -cut $C \in \mathcal{C}$ with $\sum_{a \in C \setminus \{\bar{a}\}} \bar{u}_x(a) < x(\bar{a})$, which implies $\sum_{a \in C} 1 - x(a) < 1$, contradicting our initial assumption.

To see necessity, assume x is a strictly reroutable flow and let $C \in \mathcal{C}$ be any t -separating cut. Now let $\bar{a} \in C$. Since x is strictly reroutable, there is an $\text{tail}(\bar{a})$ - t -flow of value $x(\bar{a})$ in $(V, A \setminus \{\bar{a}\})$ with capacities \bar{u}_x . By max flow/min cut this implies $\sum_{a \in C \setminus \{\bar{a}\}} \bar{u}_x(a) \geq x(\bar{a})$. \square

In the following, we identify those cuts that might violate the condition given in Lemma 12 for a (non-strictly) reroutable flow. We then show that this class of cuts forms a semi-lattice. This allows us to apply an uncrossing of the flow paths that iteratively eliminates the problematic cuts while maintaining reroutability.

Bad cuts Let x be an s - t -flow and let $C \in \mathcal{C}$ be a t -separating cut. An arc $\bar{a} \in C$ is (x, C) -bad if there is an arc $a \in C$ and a path $P \in \mathcal{P}_{\bar{a} \rightarrow a}$ with $x(P) > 0$. A cut C is x -bad if all arcs $\bar{a} \in C$ are (x, C) -bad.

Lemma 13. *Let x be a reroutable flow for capacities $u \equiv 1$. Let $C \in \mathcal{C}$ be a t -separating cut. If $\sum_{a \in C} (1 - x(a)) < 1$ then C is x -bad.*

Proof. By contradiction assume C is not x -bad. Then there must be an arc $\bar{a} \in C$ that is not (x, C) -bad. This implies that $\sum_{P \in \mathcal{P}_{\bar{a} \rightarrow a}} x(P) = 0$ for every $a \in C \setminus \{\bar{a}\}$. In particular, $\bar{u}_{x, \bar{a}}(a) = \bar{u}_x(a) = 1 - x(a)$ for all $a \in C \setminus \{\bar{a}\}$. Since all flow in the rerouting of x for failure of \bar{a} needs to cross $C \setminus \{\bar{a}\}$, we obtain $\sum_{a \in C \setminus \{\bar{a}\}} \bar{u}_{x, \bar{a}}(a) \geq x(\bar{a})$. Adding $1 - x(\bar{a})$ to both sides of this inequality yields a contradiction. \square

Lemma 14. *Let x be a flow and let $S, S' \in \mathcal{S}$ be such that $\delta^+(S)$ and $\delta^+(S')$ are both x -bad. Then $\delta^+(S \cup S')$ is an x -bad t -separating cut as well.*

Proof. Define $C := \delta^+(S)$, $C' := \delta^+(S')$, and $C^* := \delta^+(S \cup S')$. Let $\bar{a} \in C^*$. We will show that \bar{a} is (x, C^*) -bad, proving the lemma. Note that $C^* \subseteq C \cup C'$ and hence $\bar{a} \in C$ or $\bar{a} \in C'$. Without loss of generality assume the former. Because, C is x -bad, \bar{a} is (x, C) -bad. Therefore there must be an arc $a \in C$ and a path $P \in \mathcal{P}_{\bar{a} \rightarrow a}$ with $x(P) > 0$. Consider the suffix $Q := P[\text{tail}(a), t]$ of P starting with arc a . Observe that Q starts in $S \cup S'$ but ends in $t \notin S \cup S'$. Hence Q crosses C^* , i.e., there is $a' \in Q \cap C^* \subseteq P \cap C^*$. Observe that $\bar{a} \prec_P a \preceq_P a'$, i.e., $P \in \mathcal{P}_{\bar{a} \rightarrow a'}$, showing that \bar{a} is (x, C^*) -bad. \square

Uncrossing paths Let $P \in \mathcal{P}$. For two nodes $v, w \in V$ visited by P (in that order), we let $P[v, w]$ denote the subpath of path P starting at v and ending at w . Given another path $Q \in \mathcal{P}$ and an arc $a \in P \cap Q$, let $P \times_a Q$ be a simple s - t -path in the concatenation of $P[s, \text{head}(a)]$ and $Q[\text{head}(a), t]$.

Theorem 15. *Let x be a reroutable flow for capacities $u \equiv 1$. Then there is a strictly reroutable flow x' with $\text{val}(x') = \text{val}(x)$ and $x'(a) \leq x(a)$ for all $a \in A$.*

Proof. W.l.o.g., assume that x minimizes $\sum_{a \in A} x(a)$ among all reroutable flows x' with $\text{val}(x') = \text{val}(x)$ and $x'(a) \leq x(a)$ for all $a \in A$ (if this is not the case, we can replace x by the flow minimizing the total arc flow).

Define $\mathcal{S}' := \{S \in \mathcal{S} : \delta^+(S) \text{ is an } x\text{-bad cut}\}$. If $\mathcal{S}' = \emptyset$, Lemmas 12 and 13 imply that x is strictly reroutable and we are done. Thus assume $\mathcal{S}' \neq \emptyset$ and define $S^* := \bigcup_{S \in \mathcal{S}'} S$ and $C^* := \delta^+(S^*)$. Note that Lemma 14 implies $S^* \in \mathcal{S}'$, and that further, by construction, S^* defines the rightmost bad cut, i.e., no cut $\delta^+(S)$ for $S \in \mathcal{S}$ with $S \setminus S^* \neq \emptyset$ is x -bad.

Next we construct a digraph $H = (V_H, A_H)$ as follows. We let $V_H := C^*$, i.e., the nodes of H are the arcs of C^* . For every pair of distinct arcs $a, a' \in C^*$, we introduce the arc (a, a') in A_H if and only if there is a path $P \in \text{supp}(x) \cap \mathcal{P}_{a \rightarrow a'}$ such that a' is the last arc of C^* on P . Observe that, because C^* is x -bad, every node of H has an outgoing arc. Hence H contains a simple directed cycle Z . Let $a_1, \dots, a_k \in C^*$ be the arcs corresponding to the nodes of the cycle Z , and let $P_1, \dots, P_k \in \text{supp}(x)$ be paths corresponding to the arcs of Z , i.e., the paths fulfill that arc a_i is the last arc of P_i that crosses C^* , and $a_i \in P_i \cap P_{i+1}$ for each $i \in \{1, \dots, k\}$ (for ease of notation we identify i and j if $i \equiv j \pmod k$).

Now define $P'_i := P_{i+1} \times_{a_i} P_i$ for $i \in [k]$. Let $\varepsilon := \min_i x(P_i)$; see Fig. 2 for an illustration. We construct a new flow x' as follows:

$$x'(P) = \begin{cases} x(P) + \varepsilon & \text{if } P = P'_i \text{ for some } i, \\ x(P) - \varepsilon & \text{if } P = P_i \text{ for some } i, \\ x(P) & \text{otherwise.} \end{cases}$$

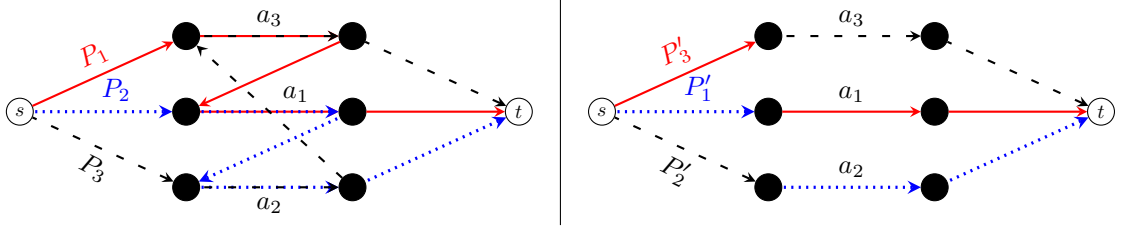


Figure 2: Uncrossing of paths on a bad cut.

We show that x' is also a reroutable flow. First observe that $\text{val}(x') = \text{val}(x)$ and that $x'(a) \leq x(a)$ for all $a \in A$. Now let $\bar{a} \in A$ and let $S \subseteq V \setminus \{t\}$ with $\text{tail}(\bar{a}) \in S$ and define $C := \delta^+(S)$. First observe that if $S \setminus S^* \neq \emptyset$, then C is not x -bad and therefore $\sum_{a \in C} (1 - x'(a)) \geq \sum_{a \in C} (1 - x(a)) \geq 1$ by Lemma 13, implying $\sum_{a \in C \setminus \{\bar{a}\}} \bar{u}_{x', \bar{a}}(a) \geq x'(\bar{a})$, and therefore x' is reroutable. Thus we consider the case $S \subseteq S^*$. We will show that in this case $\bar{u}_{x', \bar{a}}(a) \geq \bar{u}_{x, \bar{a}}(a)$ for all $a \in C$, and therefore x' is again reroutable. To this end, observe that the definition of $\bar{u}_{x, \bar{a}}$ implies

$$\begin{aligned} \bar{u}_{x', \bar{a}}(a) - \bar{u}_{x, \bar{a}}(a) &= \sum_{P \in \mathcal{P}_a \setminus \mathcal{P}_{\bar{a} \rightarrow a}} x(P) - \sum_{P \in \mathcal{P}_a \setminus \mathcal{P}_{\bar{a} \rightarrow a}} x'(P) \\ &= \varepsilon \cdot (|\underbrace{\{i : P_i \in \mathcal{P}_a \setminus \mathcal{P}_{\bar{a} \rightarrow a}\}}_{=: I}| - |\underbrace{\{i : P'_i \in \mathcal{P}_a \setminus \mathcal{P}_{\bar{a} \rightarrow a}\}}_{=: I'}|). \end{aligned}$$

We show that $i \in I'$ implies $i + 1 \in I$ and therefore $|I| \geq |I'|$, which proves our claim. Consider any $i \in I'$. We observe that $a \in P_{i+1}[s, \text{head}(a_i)]$, as $\text{tail}(a) \in S \subseteq S^*$ and no arc in $P_i[\text{head}(a_i), t]$ has its tail in S^* (recall that a_i is the last arc of P_i crossing C^*). This further implies that $P_{i+1}[s, \text{head}(a)] \subseteq P'_i[s, \text{head}(a)]$ and thus $\bar{a} \notin P_{i+1}[s, \text{head}(a)]$ as $P'_i \notin \mathcal{P}_{\bar{a} \rightarrow a}$. Therefore $P_{i+1} \in \mathcal{P}_a \setminus \mathcal{P}_{\bar{a} \rightarrow a}$, i.e., $i + 1 \in I$. Now $|I| \geq |I'|$ implies $\bar{u}_{x', \bar{a}}(a) \geq \bar{u}_{x, \bar{a}}(a)$ and hence x' is reroutable.

Finally, we show that $\sum_{a \in A} x'(a) < \sum_{a \in A} x(a)$. To this end, consider any fixed $i \in [k]$. Observe that

$$x(a_i) - x'(a_i) = \varepsilon \cdot (|\{j : a_i \in P_j\}| - |\{j : a_i \in P'_j\}|).$$

Note that $P'_j \cap C^* \subseteq P_{j+1}$ and hence $a_i \in P'_j$ implies $a_i \in P_{j+1}$, i.e., $\{j : a_i \in P'_j\} \subseteq \{j-1 : a_i \in P_j\}$. Further note that $P'_{i-1} = P_i \times_{a_{i-1}} P_{i-1}$ does not contain a_i , as $a_{i-1} \prec_{P_i} a_i$. Hence the above containment is strict and $x(a_i) - x'(a_i) > 0$.

We thus have constructed a reroutable flow x' with $\text{val}(x') = \text{val}(x)$ and $x'(a) \leq x(a)$ for all $a \in A$ and $\sum_{a \in A} x'(a) < \sum_{a \in A} x(a)$, contradicting our initial assumption. \square

Remark 16. *The proof of Theorem 15 preserves integrality. Therefore, if $x(P)$ is an integer multiple of α for every $P \in \mathcal{P}$, then x' can be chosen such that also $x'(P)$ is an integer multiple of α for every $P \in \mathcal{P}$.*

Remark 17. *The characterization of strictly reroutable flows for unit capacities given in Lemma 12 can be extended to instances with arbitrary capacities as follows: An s - t -flow x is strictly reroutable if and only if $\sum_{a \in C \setminus \{\bar{a}\}} (u(a) -$*

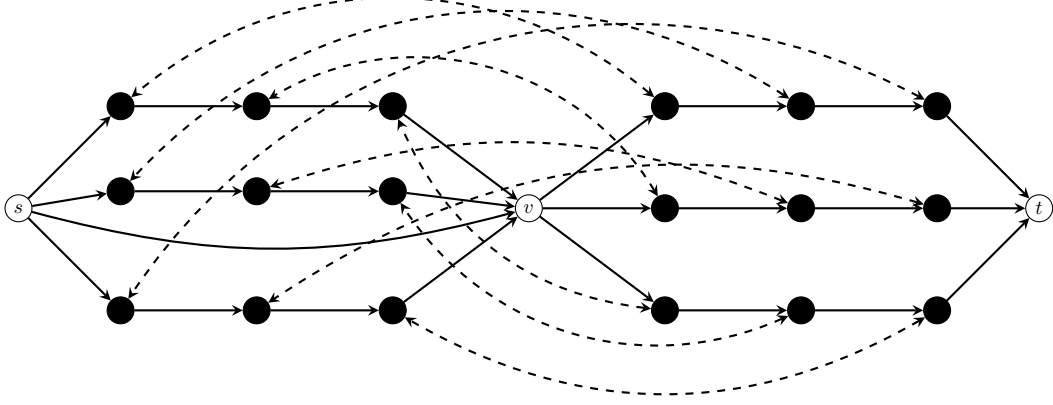


Figure 3: Example network in which no integral or half-integral reroutable flow is optimal. Dashed arcs represent bidirected backup links (see Section 2.4), all arcs have unit capacities. The maximum reroutable flow value is 2. This can only be achieved when $x(s, v) = 1$, the three s - v -paths all carry $1/3$ unit of flow, and the three v - t -paths all carry $2/3$ unit of flow. See Remark 22 for a detailed discussion.

$x(a) \geq x(\bar{a})$ for all $C \in \mathcal{C}$ and all $\bar{a} \in C$. Furthermore, if flow x violates this constraint for some cut C and an arc $\bar{a} \in C$, then \bar{a} is (x, C) -bad. However, this no longer implies that C is an x -bad cut. Indeed, consider the example given in Fig. 1 when setting $u(a_1) = 2$ and $u(a) = 1$ for all $a \in A \setminus \{a_1\}$. Let x be the reroutable (but not strictly reroutable) flow sending one unit of flow along the path (a_1, a_2, a_3) . The cut $C := \{a_1, a_3\}$ is not x -bad, but C and a_1 violate the above constraint, preventing a strict rerouting.

4 Computing (half-)integral solutions

In some application contexts, flow cannot be split into arbitrarily small pieces. This is the setting we consider in this section. We say a flow x is *integral*, if $x(P) \in \mathbb{Z}$ for all $P \in \mathcal{P}$. We say that x is *half-integral* if $2x$ is integral.

For many fundamental flow problems, such as MAX FLOW or MIN COST FLOW, integrality comes for free, i.e., as long as capacities are integral, there exists an optimal integral solution. In the case of reroutable flows, this property does not hold, see, e.g., Fig. 3. In fact, it turns out to be NP-hard to decide whether there is a non-zero integral reroutable flow in a network.

Theorem 18. *It is NP-hard to decide whether there is an integral (strictly) reroutable flow of value 1, even when restricted to instances with $u \equiv 1$.*

Note that this problem corresponds to sending a unit of flow along a single s - t -path. The hardness stems from a problem named FORBIDDEN PAIRS s - t -PATH. The reduction is described in Section 5. While it seems that Theorem 18 does not give much space for positive algorithmic results, we can do much better if we relax the integrality requirement slightly.

```

 $A_0 := \emptyset, A_1 := \emptyset$ 
while  $\exists a \in A \setminus A_0 : A_1 \cup \{a\}$  is a  $\text{tail}(a)$ - $t$ -cut in  $D$ 
     $A_0 := A_0 \cup \{a\}$ 
     $A_1 \leftarrow \{a' : a' \text{ is an } s\text{-}t\text{-bridge in } (V, A \setminus A_0)\}$ 
end while
if  $A_0$  is an  $s$ - $t$ -cut in  $D$ 
    return "No reroutable flow of value 1 exists."
else
    Let  $P_1, P_2$  be two  $s$ - $t$ -paths in  $(V, A \setminus A_0)$  such that  $P_1 \cap P_2 = A_1$ .
    Let  $x$  be the flow defined by  $x(P_1) = x(P_2) = 1/2$ .
    return  $x$ 
end if

```

Listing 1: Computing a half-integral reroutable unit demand flow

Theorem 19. *Given a network with $u \equiv 1$, the algorithm given in Listing 1 computes in polynomial time either a half-integral strictly reroutable flow of value 1, or correctly determines that no reroutable flow of value 1 exists.*

In particular, this implies that if we are interested in sending a single unit of flow, we never need to split our flow in more than two paths. Before we discuss the algorithm from Theorem 19, let us shortly discuss the case of arbitrary capacities. As a consequence of the max flow/min cut result proven in Section 2.3, we obtain the following approximation.

Theorem 20. *If u is integral, then there is a strictly reroutable half-integral flow x with $\text{val}(x) \geq \text{OPT}/2$, where OPT is the value of a maximum reroutable flow. The flow x can be computed in polynomial time.*

Proof. Recall that in the proof of Theorem 8 we computed an s - t -flow x' that was maximal with respect to capacities $u'(a) := \min\{u(a), u(C_a \setminus \{a\})\}$. We then showed that the flow $x := x'/2$ is strictly reroutable and within a factor of 2 of a corresponding R-cut. In particular, $\text{val}(x)$ is within a factor of 2 of the maximum reroutable flow value. Note that if u is integral, also u' is integral, and hence we can choose x' to be integral, ensuring that x is half-integral. \square

Algorithm for computing a half-integral flow for unit demand A natural starting point for an algorithm is to identify arcs $a \in A$ such that $\text{tail}(a)$ is disconnected from t in $(V, A \setminus \{a\})$. Obviously, no reroutable flow can send a positive amount of flow along such arcs, as after failure of a , the flow cannot be rerouted to t . Surprisingly, this simple preprocessing step can be generalized to an iterative procedure that solves the problem.

The algorithm, which is formally given in Listing 1, maintains two sets A_0 and A_1 . In every iteration, it identifies an arc that cannot carry any flow in any reroutable flow and adds it to A_0 . The set A_1 contains the s - t -bridges in the graph $(V, A \setminus A_0)$, i.e., all arcs whose removal disconnects s from t in that graph. Clearly, if $x(a) = 0$ for all $a \in A_0$, then every arc in A_1 must carry 1 unit of flow. If at some point A_0 becomes an s - t -cut, we know that no reroutable flow of value 1 exists. On the other hand, if the algorithm finds no more arcs to add

to A_0 while s and t are still connected in $(V, A \setminus A_0)$, it computes two paths P_1, P_2 that only intersect at the bridges, and sends $1/2$ units of flow along each of them.

Proof of Theorem 19. To see that Algorithm 1 terminates in polynomial time, observe that $|A_0|$ is increased in every iteration of the while-loop and the loop thus terminates after at most $|A|$ iterations, each of which can be carried out in polynomial time.

Case 1: No flow exists. We now show that if Algorithm 1 denies the existence of a reroutable flow of value 1, this is indeed correct. By contradiction assume A_0 contains an s - t -cut but there exists a reroutable flow x of value 1. We prove by induction that at any step of algorithm the set A_0 fulfills the property that $x(a) = 0$ for all $a \in A_0$, yielding a contradiction. The claim is clearly true initially, when $A_0 = \emptyset$. Now consider any iteration of the while-loop, considering arc a . By induction hypothesis, every s - t -path P with $x(P) > 0$ must be a path in $(V, A \setminus A_0)$. Note that there is an order a_1, \dots, a_ℓ of the set A_1 of s - t -bridges of $(V, A \setminus A_0)$ such that every such flow-carrying path contains all of these bridges in exactly that order. In particular $x(a_1) = \dots = x(a_\ell) = 1$. Now consider the next arc a added to A_0 and assume by contradiction that $x(a) > 0$. By choice of a there is a $\text{tail}(a)$ - t -cut $C \subseteq A_1 \cup \{a\}$ in D . Note that if $C \cap A_1 = \emptyset$, there is no rerouting of x in case of failure of arc a , as there is no $\text{tail}(a)$ - t -path in $(V, A \setminus \{a\})$. Thus, let $a_k \in C \cap A_1$ be the bridge with the highest index k on the cut. We distinguish two cases:

- (i) Assume a appears before a_k on every flow-carrying path. Note that C is a $\text{tail}(a_k)$ - t -cut because $a_k \in C$ and that $\sum_{a' \in C} \bar{u}_{x, a_k}(a') = 1 - x(a) < 1$. Therefore, the one unit of flow on a_k cannot be rerouted when a_k fails.
- (ii) Now assume a occurs after a_k on every flow-carrying path. But then, when a fails, the flow on a cannot be rerouted as all edges in $C \setminus \{a\} \subseteq A_1$ occur before a on every flow-carrying path and thus $\sum_{a' \in C \setminus \{a\}} \bar{u}_{x, a}(a') = 0$.

We thus deduce that $x(a) = 0$, completing the induction.

Case 2: Algorithm returns flow. Finally, we show that if $(V, A \setminus A_0)$ contains an s - t -path after completing the while-loop, then the flow x returned by the algorithm is a strictly reroutable flow. First observe that two s - t -paths P_1, P_2 in $(V, A \setminus A_0)$ with $P_1 \cap P_2 = A_1$ exist by the max flow/min cut theorem, as A_1 contains exactly the bridges of $(V, A \setminus A_0)$. Now consider the failure of any arc $\bar{a} \in A \setminus A_0$. Let C be a $\text{tail}(\bar{a})$ - t -cut in D minimizing $U(C) := \sum_{a \in C \setminus \{\bar{a}\}} \bar{u}_x(a)$. We show that $U(C) \geq x(\bar{a})$, which by max flow/min cut implies that there is a rerouting of x in case of failure of \bar{a} . By termination condition of the while-loop, there is at least one arc $a' \in C \setminus (A_1 \cup \{\bar{a}\})$. Note that $x(a') \in \{0, 1/2\}$ and thus $U(C) \geq 1/2$. If $\bar{a} \notin A_1$, then $x(\bar{a}) \leq 1/2 \leq U(C)$. If $\bar{a} \in A_1$, we distinguish two cases.

- (i) If $x(a') = 0$ then $U(C) \geq 1$ and the one unit of flow on \bar{a} can be rerouted.
- (ii) If $x(a') = 1/2$, then $a' \notin A_0$. Note that C is a $\text{tail}(a')$ - t -cut in D and thus there is $a'' \in C \setminus A_1 \cup \{a'\}$ by termination condition of the while-loop. Note that, because $a'' \notin A_1$, we have $a'' \neq a$ and $x(a'') \leq 1/2$. Thus $U(C) \geq 1$ also in this last case.

We conclude that x is indeed strictly reroutable. \square

Remark 21. Note that our proof of Theorem 19 does not make use of Theorem 15. Instead, it gives a simple alternative argument for the equivalence of reroutable and strictly reroutable flows in unit capacity networks, for the special case of unit value flows.

Remark 22. Theorem 19 implies that, for networks with $u \equiv 1$, if there exists any reroutable flow of value 1, then there exists a half-integral strictly reroutable flow of value 1. The example given in Fig. 3, however, reveals that this is no longer true for flows of higher value. To see this, consider any reroutable flow of value Δ in the depicted network. Recall that backup links can only be used for rerouting. Thus, all nominal flow (i.e., before failure) must pass one of the three s - v -paths or the arc (s, v) . Let x_1, x_2, x_3 be the flow values on these three paths and x^* be the flow value on the arc (s, v) . Also, all flow must pass one of the three v - t -paths. Let x'_1, x'_2, x'_3 be the arc flow values on these paths. We obtain $x_1 + x_2 + x_3 + x^* = x'_1 + x'_2 + x'_3 = \Delta$. We further show that $x_i + x'_j \leq 1$ for every i, j : To see this, consider the rerouting when the i th arc of the j th v - t -path fails. Observe that the only backup link leads to a node on the i th s - v -path. Hence $x_i + x'_j \leq 1$. We deduce that $3 + x^* \geq 2\Delta$. For $\Delta = 2$ this yields the unique solution $x^* = 1$, $x_1 = x_2 = x_3 = 1/3$, and $x'_1 = x'_2 = x'_3 = 2/3$, which can be verified to be a strictly reroutable flow. Note that the example can be generalized to arbitrarily small fractional values by introducing k instead of only 3 parallel paths in each of the two segments.

5 Hardness results

In this section, we give hardness results for MAX RF and some variants of the problem.

Paths avoiding forbidden pairs Our hardness results are based on reductions from FORBIDDEN PAIRS s - t -PATH, which is defined as follows: We are given a digraph $D' = (V', A')$, two nodes $s', t' \in V'$, and a set of forbidden arc pairs $\mathcal{F} \subseteq \{\{a, \bar{a}\} : a, \bar{a} \in A'\}$. The task is to find an s' - t' -path P that does not contain both arcs of any pair, i.e., $|S \cap P| \leq 1$ for all $S \in \mathcal{F}$. It is not hard to see that FORBIDDEN PAIRS s - t -PATH is NP-hard [14].

In all reductions that follow, we will implicitly make the following three assumptions on the digraph $D' = (V', A')$ given in the FORBIDDEN PAIRS s - t -PATH instance.

1. Forbidden pairs are disjoint, i.e., $S \cap S' = \emptyset$ for $S, S' \in \mathcal{F}$.
2. If $a \in S$ for some $S \in \mathcal{F}$, then $\delta^+(\text{tail}(a)) = \{a\}$.
3. If $a \in S$ and $a' \in S'$ for some $S, S' \in \mathcal{F}$, then $\text{head}(a) \neq \text{tail}(a')$.

It is easy to see that these assumptions are without loss of generality. They can be ensured by subdividing arcs, without changing the feasibility of the FORBIDDEN PAIRS s - t -PATH instance.

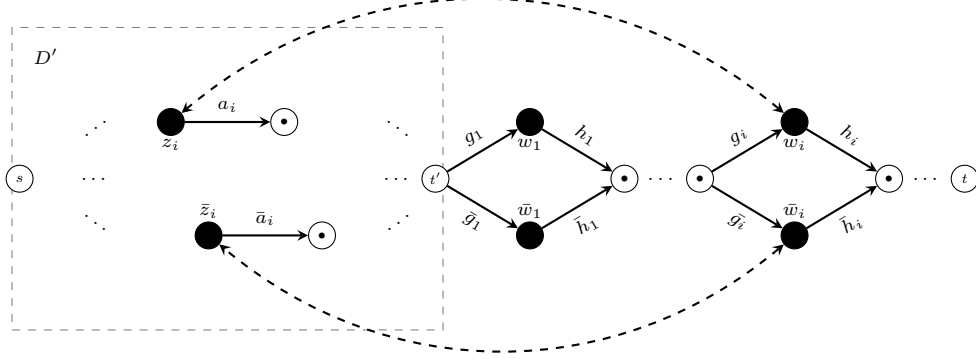


Figure 4: Construction for the proof of Theorem 4. The dashed box contains the graph D' from the FORBIDDEN PAIRS s - t -PATH instance. The arcs a_i, \bar{a}_i have capacity 2 for all i , all other arcs have unit capacity. In a reroutable flow of value 2, the arcs h_i and \bar{h}_i must be saturated for all i . Any rerouting for a_i has to traverse h_i and any rerouting for \bar{a}_i has to traverse \bar{h}_i .

5.1 General capacities

Theorem 4. MAX RF is NP-hard, even when $u(a) \in \{1, 2\}$ for all $a \in A$.

Proof. We are given an instance $(D' = (V', A'), s', t', \mathcal{F})$ of FORBIDDEN PAIRS s - t -PATH. We construct an instance of MAX RF as follows. Denote the forbidden pairs in \mathcal{F} by $\{a_1, \bar{a}_1\}, \dots, \{a_k, \bar{a}_k\}$ in some arbitrary order. We introduce new nodes v_2, \dots, v_{k+1} , w_1, \dots, w_k , $\bar{w}_1, \dots, \bar{w}_k$ and define $s := s'$, $v_1 := t'$, and $t := v_{k+1}$. We then introduce arcs $g_i = (v_i, w_i)$, $h_i = (w_i, v_{i+1})$ and $\bar{g}_i = (v_i, \bar{w}_i)$, $\bar{h}_i = (\bar{w}_i, v_{i+1})$ for every $i \in [k]$. We also add the arc (s, t') . Furthermore, we introduce backup links from v_i to t for every $i \in [k]$ and from v to t for every $v \in V'$. For every forbidden pair of arcs $\{a_i, \bar{a}_i\}$, we introduce four backup paths: from $z_i := \text{tail}(a_i)$ to w_i and vice versa and from $\bar{z}_i := \text{tail}(\bar{a}_i)$ to \bar{w}_i and vice versa. Finally, we set capacities $u(a_i) := u(\bar{a}_i) := 2$ for all $i \in [k]$ and $u(a) := 1$ for all other arcs a . See Fig. 4 for an illustration of the construction.

We show that the MAX RF instance constructed above allows for a reroutable flow of value 2 if and only if there is an s' - t' -path avoiding the forbidden pairs in \mathcal{F} . In the following, we call a node v *safe* if there is a backup link from v to t .

Sufficiency Assume there is an s' - t' -path P in D' avoiding all forbidden pairs.

We will construct a reroutable flow of value 2. First, we extend P to an s - t -path Q as follows. For $i \in [k]$, attach g_i and h_i to Q if $a_i \in P$, and attach \bar{g}_i and \bar{h}_i otherwise. As for every i , the path Q uses either h_i or \bar{h}_i , there is another s - t -path \bar{Q} in G that is arc-disjoint from Q (starting with (s, t') and then using the complement of Q from t' to t). We set $x(Q) := x(\bar{Q}) := 1$.

We now verify that x is a reroutable flow. As the nodes w_i , \bar{w}_i , z_i , and \bar{z}_i for $i \in [k]$ are the only nodes that are not safe, we only need to check that there is a rerouting for failure of the arcs a_i , \bar{a}_i , h_i and \bar{h}_i . Because

$\bar{u}_{x,h_i}(a_i) = 2 - x(a_i) \geq 1$, we can concatenate the backup link from w_i to z_i with the arc a_i and the backup link from $\text{head}(a_i)$ to t to obtain a path with residual capacity 1. This is enough to reroute the flow x in case of failure of h_i . An identical argument applies for failure of \bar{h}_i . Further note that $a_i \in Q$ implies $h_i \in Q$ by construction of Q . Therefore, either $x(a_i) = 0$ or $\bar{u}_{x,a_i}(h_i) = 1$. In the former case, rerouting is trivial. In the latter case, we can concatenate the backup link from z_i to w_i with h_i and the backup link from v_{i+1} to t to reroute the one unit of flow on a_i . Likewise, we observe that $\bar{a}_i \in Q$ implies $\bar{h}_i \in Q$, because $\bar{a}_i \in P$ implies $a_i \notin P$. Hence we can construct a rerouting for \bar{a}_i in the same manner.

Necessity Now assume there exists a reroutable flow x of value 2. Observe that $x(h_i) = x(\bar{h}_i) = 1$ for all $i \in [k]$ (as the flow cannot use the backup links and thus has to traverse these arcs). Let $Q \in \mathcal{P}$ be any path with $x(Q) > 0$ and $(s, t') \notin Q$. Let $P' := Q[s', t']$ be the projection of Q to D' . We claim that P' avoids all forbidden pairs in \mathcal{F} . By contradiction assume this is not the case, i.e., $a_i, \bar{a}_i \in P' \subseteq Q$ for some i . Consider the rerouting x_{a_i} for failure of a_i . Note that assumptions (i) and (ii) from the beginning of the proof imply that the only z_i - t -path in $(V, A \setminus \{a_i\})$ contains h_i . Hence all flow in the rerouting has to traverse h_i , and therefore

$$x(a_i) = x_{a_i}(h_i) \leq \bar{u}_{x,a_i}(h_i) = \sum_{P \in \mathcal{P}_{a_i \rightarrow h_i}} x(P) \leq x(a_i).$$

We conclude that all the above inequalities must be fulfilled with equality, and hence $Q \in \mathcal{P}_{a_i \rightarrow h_i}$, and in particular $h_i \in Q$. By a symmetric argument, we conclude that $\bar{a}_i \in Q$ implies $\bar{h}_i \in Q$. However there is no s - t -path that contains both h_i and \bar{h}_i . This yields the desired contradiction. Hence P' is a path avoiding all forbidden pairs. \square

5.2 Computing integral flows

In Section 4 we provided an algorithm that was able to decide whether a $1/2$ -integral reroutable flow of value 1 exists in a unit capacity network. If however, we require the flow to be integral, this question suddenly becomes *NP*-hard. Note that this problem corresponds to finding a single ‘reroutable’ path.

Theorem 18. *It is NP-hard to decide whether there is an integral (strictly) reroutable flow of value 1, even when restricted to instances with $u \equiv 1$.*

Proof. Let $(D' = (V', A'), s', t', \mathcal{F})$ be an instance of FORBIDDEN PAIRS s - t -PATH. We construct an instance of MAX RF as follows. We let $s := s'$ and $t := t'$. For every forbidden pair $\{a, \bar{a}\} \in \mathcal{F}$, we add a backup path from $\text{tail}(a)$ to $\text{tail}(\bar{a})$ and vice versa. For every node v that is not the tail of an arc that appears in a forbidden pair, we add a backup link to t . We set unit capacities to all arcs.

Let P be an s - t -path avoiding all forbidden pairs. Then the flow x with $x(P) = 1$ is a reroutable flow. To see this, observe that the only nodes that are not safe are the tails of arcs participating in forbidden pairs. Consider a forbidden pair $\{a, \bar{a}\} \in \mathcal{F}$. Observe that $x(a) = 0$ or $x(\bar{a}) = 0$ and hence a rerouting exists in case of failure of either of the two arcs.

Now assume there is an integral reroutable flow of value 1. This means that $x(P) = 1$ for an s - t -path P that does not use any backup link. Assume that $a, \bar{a} \in P$ for some $\{a, \bar{a}\} \in \mathcal{F}$. W.l.o.g. assume $a <_P \bar{a}$. Note that $\{a, \bar{a}\}$ is a $\text{tail}(\bar{a})$ - t -cut in D . Hence, when \bar{a} fails, the rerouting must use a , i. e., $x_{\bar{a}}(a) = 1 > 0 = \bar{u}_{x, \bar{a}}(a)$, a contradiction.

Because of Theorem 15 and Remark 16 the constructed instance allows for an integral strictly reroutable flow of value 1 if and only if it allows for a reroutable flow of value 1. Hence the hardness result carries over to MAX SRF. \square

5.3 Multiple arc failures

A natural generalization of MAX RF and MAX SRF allows multiple simultaneous arc failures. When a set of arcs S fails, flow is interrupted where it first encounters an arc from S and has to be rerouted from that point to the sink.

To formalize this, we introduce the following notation. For $S \subseteq A$ and $a \in A$ we define $\mathcal{P}_S := \bigcup_{\bar{a} \in S} \mathcal{P}_{\bar{a}}$ and $\mathcal{P}_{S \rightarrow a} := \bigcup_{\bar{a} \in S} \mathcal{P}_{\bar{a} \rightarrow a}$. Let x be an s - t -flow. We extend the notion of capacity available after failure of S by defining

$$\bar{u}_{x, S}(a) := u(a) - \sum_{P \in \mathcal{P}_S \setminus \mathcal{P}_{S \rightarrow a}} x(P).$$

A rerouting of x for failure of S consists of a collection of flows $(x_{S, \bar{a}})_{\bar{a} \in S}$ in $(V, A \setminus S)$, such that $x_{S, \bar{a}}$ is a $\text{tail}(\bar{a})$ - t -flow for each $\bar{a} \in S$, with

$$\text{val}(x_{S, \bar{a}}) = \sum_{P \in \mathcal{P}_{\bar{a}} \setminus \mathcal{P}_{S \rightarrow \bar{a}}} x(P) \quad \text{and} \quad \sum_{\bar{a} \in S} x_{S, \bar{a}}(a) \leq \bar{u}_{x, S}(a) \text{ for all } a \in A \setminus S.$$

A rerouting is strict if $\sum_{\bar{a} \in S} x_{S, \bar{a}}(a) \leq \bar{u}_x(a)$ for all $a \in A$. A flow is (strictly) k -reroutable if it has a (strict) rerouting for failure of any set $S \subseteq A$ with $|S| \leq k$. We denote the corresponding problem of finding a (strictly) k -reroutable flow of maximum value by MAX (STRICTLY) k -REROUTABLE FLOW. It turns out that dealing even with only 2 arc failures in unit capacity networks is NP-hard in both cases.

Theorem 25. MAX (STRICTLY) k -REROUTABLE FLOW is NP-hard, even when restricted to instances with $k = 2$ and $u \equiv 1$.

Proof. Let $(D' = (V', A'), s', t', \mathcal{F})$ be an instance of FORBIDDEN PAIRS s - t -PATH. We construct an instance $(D = (V, A), s, t, u)$ of MAX 2-REROUTABLE FLOW as follows. Let $\{a_1, \bar{a}_1\}, \dots, \{a_\ell, \bar{a}_\ell\}$ be the forbidden pairs in \mathcal{F} in some arbitrary order. For $i \in [\ell]$ we add vertices v_i, w_i, v'_i, w'_i and $\bar{w}_i, \bar{v}'_i, \bar{w}'_i$, and define $v_{\ell+1} := s'$. We add arcs $g_i = (v_i, w_i), h_i = (w_i, v'_i), g'_i = (v'_i, w'_i), h'_i = (w'_i, v_{i+1})$ as well as $\bar{g}_i = (v_i, \bar{w}_i), \bar{h}_i = (\bar{w}_i, \bar{v}'_i), \bar{g}'_i = (\bar{v}'_i, \bar{w}'_i), \bar{h}'_i = (\bar{w}'_i, v_{i+1})$. We generalize the concept of backup links to 2-backup links, which are simply two parallel internally node-disjoint backup links. We add 2-backup links from each of v_i, v'_i , and \bar{v}'_i to t' , thus making them safe nodes. We further add 2-backup links w_i to w'_i , from w'_i to z_i , and from z_i to w_i , as well as from \bar{w}_i to \bar{w}'_i , from \bar{w}'_i to \bar{z}_i , and from \bar{z}_i to \bar{w}_i . For every node $v \in V'$ that is not the tail of an arc that participates in a forbidden pair, we add a 2-backup link from v to t . We also add an additional s' - t' -path \bar{P} of length 6 consisting of the arcs e_1, \dots, e_6 . We add 2-backup links from $\text{tail}(e_2)$ to $\text{tail}(e_4)$, from $\text{tail}(e_4)$ to $\text{tail}(e_6)$, and from

$\text{tail}(e_6)$ to $\text{tail}(e_2)$. We also add 2-backup links from each of $\text{tail}(e_1)$, $\text{tail}(e_3)$ and $\text{tail}(e_5)$ to t' . Finally, we define $s := v_1$, $t := t'$, and $u \equiv 1$.

We show two implications: (i) If there is an s' - t' -path in D' avoiding all pairs, then there is a strictly reroutable flow of value 1 in the constructed network D . (ii) If there is a reroutable flow of value 1 in the constructed network D , then there is an s' - t' -path in D' avoiding all pairs. Note that (i) and (ii) together also imply that a strictly reroutable flow of value 1 exists in D if and only if there is a reroutable flow of value 1. Therefore, both MAXIMUM k -REROUTABLE FLOW and MAXIMUM STRICTLY k -REROUTABLE FLOW can be used to solve FORBIDDEN PAIRS s - t -PATH.

- (i) Let P' be an s' - t' path in D' avoiding all forbidden pairs. Let O be the s - s' -path defined by the following rule: For $i \in [\ell]$, if $a_i \in P'$, then use the segment g_i, h_i, g'_i, h'_i , otherwise use the segment $\bar{g}_i, \bar{h}_i, \bar{g}'_i, \bar{h}'_i$. Let Q be the concatenation of O and P' to an s - t -path. Further, let \bar{O} be the s - s' -path that uses the segments not used by O and let \bar{Q} be the concatenation of \bar{O} and \bar{Q} . We define the flow $x(Q) := 1/2$ and $x(\bar{Q}) := 1/2$.

Consider the failure of any set $S \subseteq A$ with $|S| = 2$. Note that $\bar{u}_x(a) \geq 1/2$ for all $a \in A$. It is also easy to see that for any $\bar{a} \in A$ there are three disjoint $\text{tail}(\bar{a})$ - t -paths in D and hence at least one such path in $(V, A \setminus S)$. If $Q \cap S = \emptyset$ or $\bar{Q} \cap S = \emptyset$, we have to reroute at most $1/2$ unit of flow, which is possible by the above observations. Hence, consider the case that $S = \{q_1, q_2\}$ with $q_1 \in Q$ and $q_2 \in \bar{Q}$. If there is no rerouting for S , then there must be an arc q_3 such that $\{q_1, q_2, q_3\}$ separate $\text{tail}(q_1)$ and $\text{tail}(q_2)$ from t . It is easy to see that this is only possible if $\{q_1, q_2, q_3\} = \{a_i, h_i, h'_i\}$ or $\{q_1, q_2, q_3\} = \{\bar{a}_i, \bar{h}_i, \bar{h}'_i\}$ for some $i \in [\ell]$ —w.l.o.g., we assume the former. Note that $a_i \notin \bar{Q}$ and that h_i, h'_i are both on the same path. But $h_i, h'_i \in \bar{Q}$ implies $a_i \notin Q$ by construction of Q and the fact that P' avoids forbidden pairs. Hence there must be a rerouting for x .

- (ii) Let x be a reroutable flow of value 1. We first observe that for every $i \in [\ell]$, $x(h_i) = x(h'_i) = x(\bar{h}_i) = x(\bar{h}'_i) = 1/2$. To see this, note that the nominal flow cannot use backup links and hence $x(h_i) + x(h'_i) = 1$. Now assume that $x(h_i) > 1/2$. Then consider the failure of $S_i := \{h'_i, a_i\}$. Note that all flow on h'_i and a_i must be rerouted via h_i in this case, i.e.,

$$\sum_{P \in \mathcal{P}_{S_i}} x(P) \leq \bar{u}_{x, S_i}(h_i) = 1 - x(h_i). \quad (1)$$

But $x(h'_i) = x(h_i) > 1/2 > \bar{u}_{x, \{h'_i, a_i\}}(a)$, a contradiction. Analogously, we can see that $x(\bar{h}_i) > 1/2$ is not possible. By similar arguments, namely the failure of $\{e_4, e_6\}$ we can observe that $x(e_1) = \dots = x(e_6) \leq 1/2$. Hence there must be an s - t -path Q in D with $x(Q) > 0$ and $e_1, \dots, e_6 \notin Q$. Consider the suffix $P' := Q[s', t']$. By contradiction assume P' contains a forbidden pair of arcs a_i, \bar{a}_i for some $i \in [\ell]$. Again consider the failure of $\{h'_i, a_i\}$. By (1), we obtain $\sum_{P \in \mathcal{P}_{S_i}} x(P) \leq 1/2$. Since $x(h'_i) = 1/2$, this implies that $a_i \in P$ implies $h'_i \in P$ for all P with $x(P) > 0$. In particular, $h'_i \in Q$. By a symmetric argument we obtain $\bar{h}'_i \in Q$. But $h'_i, \bar{h}'_i \in Q$ is not possible by construction of the digraph D . Thus we arrived at the desired contradiction. \square

Acknowledgments We thank David Adjiashvili and Marco Senatore for helpful discussions. This work was supported by the Alexander von Humboldt Foundation with funds of the German Federal Ministry of Education and Research (BMBF) and by an NSERC Discovery Grant.

References

- [1] David Adjiashvili, Gianpaolo Oriolo, and Marco Senatore. The online replacement path problem. In *Algorithms – ESA 2013*, volume 8125 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2013.
- [2] Charu C. Aggarwal and James B. Orlin. On multiroute maximum flows in networks. *Networks*, 39(1):43–52, 2002.
- [3] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network flows: theory, algorithms, and applications*. Prentice Hall, 1993.
- [4] Y. P. Aneja, R. Chandrasekaran, and K. P. K. Nair. Maximizing residual flow under an arc destruction. *Networks*, 38(4):194–198, 2001.
- [5] Georg Baier, Thomas Erlebach, Alexander Hall, Ekkehard Köhler, Heiko Schilling, and Martin Skutella. Length-bounded cuts and flows. In *Automata, Languages, and Programming*, volume 4051 of *Lecture Notes in Computer Science*, pages 679–690. Springer, 2006.
- [6] Dimitris Bertsimas, Ebrahim Nasrabadi, and James B. Orlin. On the power of randomization in network interdiction. *Operations Research Letters*, 44(1):114–120, 2016.
- [7] Dimitris Bertsimas, Ebrahim Nasrabadi, and Sebastian Stiller. Robust and adaptive network flows. *Operations Research*, 61:1218–1242, 2013.
- [8] Graham Brightwell, Gianpaolo Oriolo, and F. Bruce Shepherd. Reserving resilient capacity in a network. *SIAM Journal on Discrete Mathematics*, 14(4):524–539, 2001.
- [9] Chandra Chekuri, Anupam Gupta, Amit Kumar, Joseph Naor, and Danny Raz. Building edge-failure resilient networks. *Algorithmica*, 43(1-2):17–41, 2005.
- [10] Stephen R. Chestnut and Rico Zenklusen. Hardness and approximation for network flow interdiction. *Networks*, 2017.
- [11] Amaro de Sousa and Gil Soares. Improving load balance and minimizing service disruption on ethernet networks with IEEE 802.1 S MSTP. In *Workshop on IP QoS and Traffic Control*, pages 25–35, 2007.
- [12] Yann Disser and Jannik Matuschke. The complexity of computing a robust flow. Technical Report arXiv:1704.08241, 2017.
- [13] Lester R. Ford and Delbert R. Fulkerson. *Flows in networks*. Princeton Univ. Press, 1962.

- [14] Harold N. Gabow, Shachindra N. Maheshwari, and Leon J. Osterweil. On two problems in the generation of program test paths. *IEEE Transactions on Software Engineering*, SE-2(3):227–231, 1976.
- [15] Corinna Gottschalk, Arie M.C.A. Koster, Frauke Liers, Britta Peis, Daniel Schmand, and Andreas Wierz. Robust flows over time: Models and complexity. Technical report, arXiv:1608.06520, 2016.
- [16] Fabrizio Grandoni, Gaia Nicosia, Gianpaolo Oriolo, and Laura Sanità. Stable routing under the spanning tree protocol. *Operations Research Letters*, 38(5):399–404, 2010.
- [17] Alan J. Hoffman. A generalization of max flow—min cut. *Mathematical Programming*, 6(1):352–359, 1974.
- [18] Tom Leighton and Satish Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of the ACM*, 46(6):787–832, 1999.
- [19] Jannik Matuschke, S. Thomas McCormick, Gianpaolo Oriolo, Britta Peis, and Martin Skutella. Protection of flows under targeted attacks. *Operations Research Letters*, 45(1):53–59, 2017.
- [20] Robert M. Metcalfe and David R. Boggs. Ethernet: Distributed packet switching for local computer networks. *Communications of the ACM*, 19(7):395–404, 1976.
- [21] Edward Minieka. Maximal, lexicographic, and dynamic network flows. *Operations Research*, 21(2):517–527, 1973.
- [22] Steven J. Phillips and Jeffery R. Westbrook. Approximation algorithms for restoration capacity planning. In *Algorithms – ESA ’99*, volume 1643 of *Lecture Notes in Computer Science*, pages 101–115. Springer, 1999.
- [23] F. Bruce Shepherd. Single-sink multicommodity flow with side constraints. In *Research Trends in Combinatorial Optimization*, pages 429–450. Springer, 2009.